In the Claims

1. (Cancelled)

2. (Currently Amended) A method according to claim 24 wherein the first <u>component</u> ~~data comprises compressed components which are~~<u>is</u> decompressed when copied to the executable memory.

3. (Currently Amended) A method according to claim 24 wherein the first ~~data file~~ ~~is~~<u>component is</u> decompressed as a whole when copied to the executable memory.

4. (Cancelled)

5. (Currently Amended) A method according to claim 24 wherein the further ~~data file~~ <u>component</u> comprises compressed data which is decompressed when selectively copied to the executable memory.

6. (Amended) A method according to claim 24 wherein the further ~~data file~~<u>component</u> is decompressed component by component when selectively copied to the executable memory.

7. (Amended) A method according to claim 24 wherein the first ~~data file~~<u>component</u> comprises core operating system data for the computing device.

8. (Original) A method according to claim 7 wherein the core operating system data comprises program code for enabling boot-up of the computing device and access to read only file system (ROFS) data for the computing device.

9. (Currently Amended) A method according to claim 8 wherein the first ~~data file~~<u>component</u> further comprises selected components of the ROFS data.

10. (Currently Amended) A method according to claim 24 wherein the further ~~data file~~component comprises ROFS data.

11. (Currently Amended) A method according to claim 10 wherein the further ~~data file~~component comprises an executable program.

12. (Currently Amended) A method according to claim 10 or 11 wherein the further ~~data file~~component comprises a dynamic link library.

13. (Currently Amended) A method according to claim 24 wherein the first ~~data file~~component is in the form of one or more ROM images.

14. (Currently Amended) A method according to claim 24 wherein the location of at least one of the first and further ~~data file~~components within the non-executable memory is determined by reading an address from a section of the non-executable memory.

15. (Currently Amended) A method according to ~~any~~ claim 24 wherein additional data is selectively copied to the executable memory in addition to the first and further ~~data file~~components in the composite file system.

16. (Currently Amended) A method according to claim 15 wherein the additional data is selectively copied to the ~~composite data file~~single file system.

17. (Previously Presented) A method according to claim 15 or 16 wherein the additional data comprises a language pack image.

18. (Currently Amended) A method according to claims 15 or 16 wherein a common driver is used to selectively copy the first and further ~~data file~~components to the executable memory.

19. (Currently Amended) A method according to claims 15 or 16 wherein the first and further ~~data file~~components, and the additional data, are stored in a section of the non-executable memory locked to a user.

20. (Previously Presented) A method according to claim 24 wherein the non-executable memory is selected to comprise NAND flash memory.

21. (Previously Presented) A method according to claim 24 wherein the executable memory is selected to comprise random access memory (RAM).

22. (Currently Amended) A computing device comprising a non-executable memory, an executable memory, and a file server, and programmed to ~~operate according to a method as claimed in claim 24~~:

    store an operating system, comprising a first component and a further component, in the non-executable memory;

    copy the first component from the non-executable memory to the executable memory at operating system start-up, so as to permanently shadow the first component in the executable memory during device operation;

    present, to the file server, the first component in the executable memory and the further component in the non-executable memory as a single file system;

    wherein the first component is accessed from the executable memory, where it is shadowed permanently, and the further component is accessed by selectively copying the further component to the executable memory upon demand by the file server, for subsequent use with the permanently shadowed first component by the file server.

23. (Amended) Computer software arranged to cause a computing device, <u>comprising a non-executable memory, an executable memory, and a file server,</u> to ~~operate according to a method as claimed in claim 24~~<u>:</u>

<u>store an operating system, comprising a first component and a further component, in the non-executable memory;</u>

<u>copy the first component from the non-executable memory to the executable memory at operating system start-up, so as to permanently shadow the first component in the executable memory during device operation;</u>

<u>present, to the file server, the first component in the executable memory and the further component in the non-executable memory as a single file system;</u>

<u>wherein the first component is accessed from the executable memory, where it is shadowed permanently, and the further component is accessed by selectively copying the further component to the executable memory upon demand by the file server, for subsequent use with the permanently shadowed first component by the file server.</u>

24. (Currently Amended) A method of operating a computing device, the method comprising:

storing <u>an operating system, comprising a</u> first <u>component</u> and <u>a</u> further ~~data files~~<u>component,</u> in a non-executable memory of the computing device;

copying the first ~~data file~~<u>component</u> from the non-executable memory of the computing device to executable memory <u>at operating system start-up, so as to permanently shadow the first component in the executable memory during device operation;</u>

presenting<u>, to a file server of the computing device,</u> the first ~~data file~~<u>component</u> in executable memory and the further ~~data file~~<u>component</u> in non-executable memory as a <u>single</u> ~~n executable composite~~ file system;

wherein the first ~~data file~~<u>component</u> is accessed from the executable memory, <u>where it is shadowed permanently,</u> and the further ~~data file~~<u>component</u> is accessed by selectively copying ~~one or more components of~~ the further ~~data file~~<u>component</u> to the executable memory <u>upon demand by the file server, for subsequent use with the permanently shadowed first component by the file server.</u>